PATH SYNTHESIS OF FOUR-BAR MECHANISM USING HARMONY SEARCH OPTIMIZATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR DEGREE OF
Bachelor of Technology
in
Mechanical Engineering

By
AHMED SAEED MOHSEN ALHAJJ
Roll No: 1O7ME063

Department of Mechanical Engineering
National Institute of Technology
Rourkela
2010-2011

PATH SYNTHESIS OF FOUR-BAR MECHANISM USING HARMONY SEARCH OPTIMIZATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR DEGREE OF
Bachelor of Technology
in
Mechanical Engineering

By

AHMED SAEED MOHSEN ALHAJJ
Roll No: 1O7ME063

Under the Guidance of
Prof. J.Srinivas

Department of Mechanical Engineering
National Institute of Technology
Rourkela
2010-2011

National Institute of Technology
ROURKELA

CERTIFICATE

This is to certify that the thesis entitled, "**PATH SYNTHESIS OF FOUR-BAR MECHANISM USING HARMONY SEARCH OPTIMIZATION"** submitted by **Ahmed Saeed Mohsen Al-hajj** in partial fulfilment of the requirements for the award of Bachelor of Technology in **Mechanical Engineering** at the National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any Degree or Diploma.

Date:

Prof. J.Srinivas

Dept. of Mechanical Engg.
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

I avail this opportunity to extend my hearty indebtedness to my guide **Prof J.Srinivas,** Mechanical Engineering Department, for his valuable guidance, constant encouragement and kind help at different stages for the execution of this dissertation work.

I also express my sincere gratitude to him for extending their help in completing this project.

I take this opportunity to express my sincere thanks to my project guide for co-operation and to reach a satisfactory conclusion.

AHMED SAEED MOHSEN ALHJAJJ

Roll No: 1O7ME063

Mechanical Engineering

National Institute of Technology

Rourkela

ABSTRACT

The study of four-bar linkage to trace a desired path is an important part of obtaining the optimal geometry of a four-bar linkage which is used in design. When the number of the precision points exceeds a certain number, it is not possible to apply or it is difficult to apply analytical methods. Alternatively some intelligence optimization methods can be used based on the complexity of the problem. Path synthesis of four-bar linkages with multiple number of precision points has been illustrated as an optimization problem. The objective to be minimized is sum square error of all the points reached by the coupler point. The important constraints included are 1) Grashof's condition to have a crank rocker linkage,2) the transmission angle criterion and 3) input link angle sequence. The variables included are the link lengths and angles. A new optimization scheme known as harmony search method is implemented to get the optimum solutions. Harmony search method is based on musical performance process that occurs when a musician searches for a better state of harmony. The pitch of each musical instrument determines aesthetic quality just as objective function value assigned to variables. A computer programme based on this algorithm is implemented in MATLAB to obtain the optimum dimension of four-bar linkage. The function files are employed to generate initial feasible solutions and effective object function based on penalty function method. Four examples are illustrated to show the effectiveness of the method. It is found that the method is quite convenient and works in-par with other non conventional optimization methods like genetic algorithms and particle swam optimization.

CONTENTS

Page No

Chapter 4 RESULTS AND DISCUSSION

Chapter 5 CONCLUSIONS

CHAPTER-1

INTRODUCTION

Many machine design problems require creation of a device with particular motion characteristics. Synthesis of mechanism refers to design a linkage for a prescribed motion or path or velocity of tracing joint or link there are types of synthesis technique available in literature. The following methods of synthesis are commonly found in literature: (i) Qualitative synthesis, which is a creation of potential solution in the absence of an algorithm that configures or predicts solution, (ii) type synthesis, which is a definition of proper type of mechanism best suited to the problem and is a form of qualitative synthesis and (iii) dimensional synthesis, referring to the determination of lengths of links necessary to accomplish the desired motion.

1.1 TYPE, NUMBER AND DIMENSION SYNTHESIS

Type synthesis refers to the kind of mechanism selected; it might be a linkage, a geared system, belts and pulleys, or even a cam system. This beginning phase of the total design problem usually involves design factors such as manufacturing processes, materials, safety, space, and economics. The study of kinematics is usually only slightly involved in type synthesis. Number synthesis deals with the number of links and the number of joints or pairs that are required to obtain certain mobility. Number synthesis is the second step in the design. The third step in design namely determining the dimensions of the individual links, is called dimensional synthesis.

Following are various problems occurring in dimensional synthesis.

(1) Function Generation

A frequent requirement in design is that of causing an output member to rotate, oscillate, or reciprocate according to a specified function of time or function of the input motion. This is called function generation. That is correlation of an input motion with an output motion in a linkage. A simple example is that of synthesizing a four-bar linkage to generate the function the function y=f(x). In this case, x would represent the motion (crank angle) of the input crank, and the linkage would be designed so that the motion (angle) of the output rocker would approximate the function y. Other examples of function generation are as follows: In a conveyor line the output member of a mechanism must move at the constant velocity of the conveyor while performing some operation for example, bottle capping, return, pick up the cap, and repeat the operation. The output member must pause or stop during its motion cycle to provide time for another event. The second event might be a sealing, stapling, or fastening operation of some kind. The output member must rotate at a specified no uniform velocity function because it is geared to another mechanism that requires such a rotating motion.

2. Path generation

A second type of synthesis problem is called path generation. This refers to a problem in which a coupler point is to generate a path having a prescribed shape that is controlling a point in a plane such that it follows some prescribed path. Common requirements are that a portion of the path be a circular arc, elliptical, or a straight line. Sometimes it is required that the path cross over itself. For this minimum 4-bar linkage are needed. It is commonly to arrive a point at a particular location along the path without/with prescribed times.

3. Motion Generation

The third general class of synthesis problem is called body guidance. Here we are interested in moving an object from one position to another. The problem may call for a simple translation or a combination of translation and rotation. In the construction industry, for example, heavy parts such as a scoops and bulldozer blades must be moved through a series of prescribed positions.

4. Hybrid Task synthesis

Certain applications may not be represented by a single task. It is conceivable that a task may require an object to be moved along a trajectory on which the orientation of the object may be important at a few points, while restriction on orientation could be relaxed at others. Furthermore, the task may require that a functional input/output relation exists at a few points along the trajectory. This scenario calls for hybrid task synthesis. The main benefit from a mechanism that performs a hybrid task is that the entire motion cycle becomes active, that is, during a single crank rotation the same mechanism can be used to perform several subtasks simultaneously. For example, the task may dedicate a portion of the trajectory to advancing an object along a path, another portion to moving an object from one conveyor to another through several positions while maintaining a desired orientation, and yet another segment to generating a functional relationship between the drive and follower links, as each may actuate valves that dispense prescribed amounts of different materials in a single package.

1.2 GRAPHICAL AND ANALYTICAL SYNTHESIS

Path generation is a subset of motion generation problem. Path generation of linkage is relatively an important problem in robotics and electronics industry. In path generation, the points prescribed for successive locations of coupler link in the plane are known as precision points. The number of precision points which can be synthesized is limited by the number of equation available for solution. Four-bar linkage can be synthesized by closed-form method up to 5 precision points for path generation with prescribed timing. Basic path synthesis problem starts with two prescribed points. There are both graphical and analytical methods available for path generation problem. Available graphical and analytical techniques are briefly explained in this section.

(A) Graphical Synthesis

Consider a four-bar linkage design in which link AB moves from A_1B_1 to A_2B_2 (Fig 1.1). To handle the problem graphically, draw the link AB in its two positions A_1B_1 and A_2B_2.

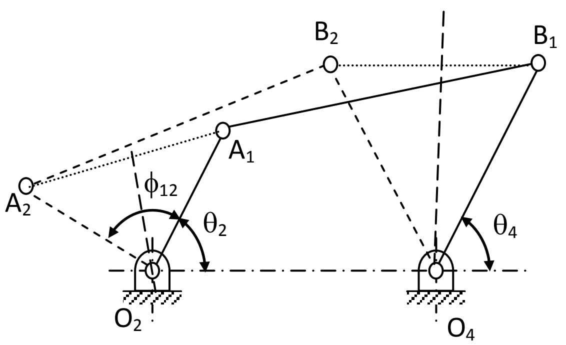


Fig.1.1 Two position synthesis (coupler motion)

Draw construction lines from A_1 to A_2 and from B_1 to B_2. Bisect the line A_1A_2 and B_1B_2 and extend the perpendicular bisector in convenient directions. Select a convenient point on each bisector as fixed pivots O_2 & O_4. Connect O_2 with A_1 and call it as link 2 and connect O_4 with B_1 and call it as link 4. The line A_1B_1 is link-3, while O_2O_4 is link-1. Check the Grashof's condition and repeat above steps if not satisfied. The graphical procedure employed for the two-position synthesis problem can be extended to the three position synthesis. As the number of precision points to be traced increases, the graphical method fails to give a correct solution.

(B) Analytical Synthesis

Synthesis for two or three precision points with analytical technique is relatively straight forward and each of these can be reduced to a system of linear simultaneous equations which are solved easily on a calculator. The four or more precision point synthesis involve the solution of nonlinear, simultaneous system of equations and so are more complicated to solve and require a computer. Fig 1.2 shows two precision-point case. In Fig 1.2, the coupler point is changing from P_1 to P_2.

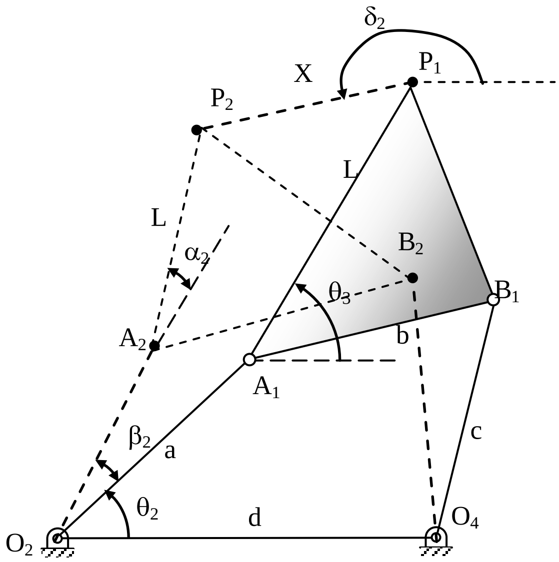


Fig.1.2 Two-position synthesis (vector loop $O_2A_2P_2P_1A_1O_2$)

As the input link O_2A_1 transverses by an angle β_2 with respect to initial position and considering the vector loop in CW direction, we get:

$$\overrightarrow{O2A2} + \overrightarrow{A2P2} = \overrightarrow{P1P2} + \overrightarrow{A1P1} + \overrightarrow{O2A1} \tag{1.1}$$

Writing in terms of link lengths and angles, we get:

$$Ae^{j(\theta 2+\beta 2)}+Le^{j(\theta 3+\alpha 2)} = Xe^{j\delta 2}+ae^{j\theta 2} +Le^{j\theta 3} \tag{1.2}$$

These result in two scalar equation in term of eight variables θ_2,β_2, a,L,θ_3,α_2, δ_2 and X . Of these, α_2, X and δ_2 are defined, while θ_2, θ_3, β_2 can be considered as a free choices (known values) and a and L are predicted. Likewise, the right side vector loop gives c and B_1P_1. There are infinite possible solutions for this problem as we may choose any set of values for the three free choices of variables in this two position case.

In this way, there are 12 variables and four equations in 3 point-synthesis. Six of them are defined in the problem statement, 2 can be chosen as free choice. As no of given positions increases, variable solutions become finite. However, the degree of complexity increases.

1.3 COUPLER CURVES

Coupler curves are used to generate useful path motions for design problems. They can approximate straight line, circular arc etc. Coupler curve is a solution to a path generation problem. It is a very useful device. The four-bar linkage has a coupler curve equation of degree 6 while slider crank linkage has a coupler curve of degree 4. Horns and Nelson atlas of four-bar coupler curve is useful reference to provide a starting point for design and analysis. It contains 7000 coupler curves and defines the linkage geometry for each of its Grashof's crank-rocker linkages. Basic method of obtaining the equation of coupler for four-bar linkage is briefly presented. A tracing point on coupler link has a coordinates (x,y) obtained by rotating crank of the linkage as shown in Fig.1.3.

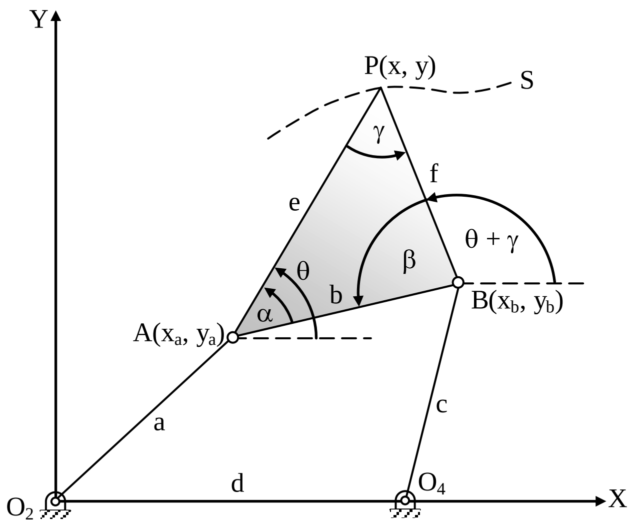


Fig.1.3 Coupler curve generated by point P

The x and y coordinates of points A and B are

$$x_a = x - e\cos\theta,\quad x_b = x - f\cos(\theta + \gamma)$$

$$y_a = y - e\sin\theta,\quad y_b = y - f\sin(\theta+\gamma) \qquad (1.3)$$

Since the points A and B describe circles with centres O_2 and O_4, we have

$$x_a^2 + y_a^2 = a^2 \text{ and}$$

$$(x_b - d)^2 + y^2 = c^2 \tag{1.4}$$

Substituting the values for (x_a, y_a) and (x_b, y_b) from Eq. (1.3), equations (1.4) become

$$[x - e\cos\theta]^2 + [y - e\sin\theta]^2 = a^2$$

$$[x - f\cos(\theta + \gamma) - d]^2 + [y - f\sin(\theta + \gamma)]^2 = c^2 \tag{1.5}$$

Simplifying and rearranging the above equations, they take following form:

$$x\cos\theta + y\sin\theta = \frac{x^2 + y^2 + e^2 - a^2}{2e} \tag{1.6}$$

$$[(x - d)\cos\gamma + y\sin\gamma]\cos\theta - [(x - d)\sin\gamma - y\cos\gamma]\sin\theta = \frac{(x-d)^2 + y^2 + f^2 - c^2}{2f} \tag{1.7}$$

Solving above equations for $\sin\theta$ and $\cos\theta$ and substituting them in the trigonometric identity

$$\sin^2\theta + \cos^2\theta = 1 \tag{1.8}$$

it results in

$$\{\sin\alpha[(x - d)\sin\gamma - y\cos\gamma](x2 + y^2 + e^2 - a^2) + y\sin\beta\,[(x - d)^2 + y2 + f^2 - c^2]\}^2$$

$$+ \{\sin\alpha\,[(x - d)\cos\gamma + y\sin\gamma](x^2 + y2 + e^2 - a^2) - x\sin\beta[(x-d)^2 + y^2 + f^2 - c^2]\}^2$$

$$= 4k^2\sin^2\alpha\,\sin^2\beta\,\sin^2\gamma\,[x(x - d) + y^2 - dy\cot\gamma]^2 \tag{1.9}$$

where $$k = \frac{e}{\sin\beta} = \frac{f}{\sin\alpha} = \frac{b}{\sin\gamma} \tag{1.10}$$

This equation describes the coupler curve of a four-link mechanism, which is of the sixth order.

The shape and state of the coupler curve changes with the changing positions of the coupler point P. Some of the important characteristics of these curves are known as double points, cusps and symmetry.

The coupler curve is said to have double points when the coupler point P passes through the same position twice as shown in Fig. 1.4. A cusp is a special case of a double point on the coupler curve, when it coincides with the instantaneous centre of the coupler in that position. Thus, when the coupler rotates about a cusp and the coupler point changes its direction of motion. The velocity of the coupler point at cusp becomes zero. There is a discontinuity in the slope of coupler curve at a cusp.

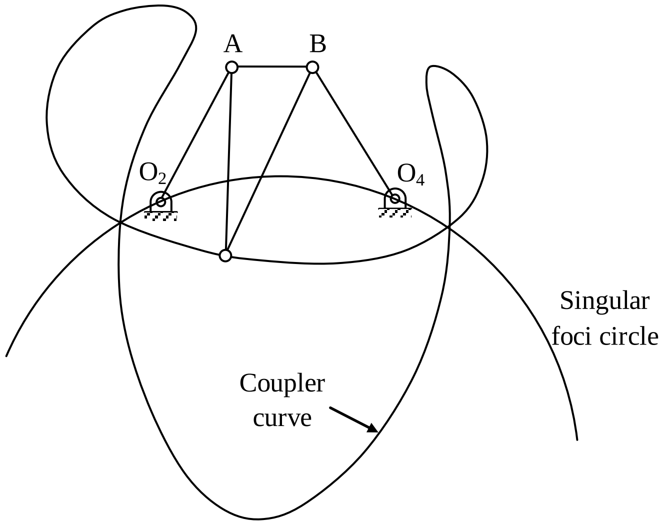


Fig. 1.4 Double Points of a Coupler Curve

The applications of the coupler curves of a four-bar linkage are found in movie camera projector's film advancement mechanism, automobile suspension systems etc.

1.4 COGNATES AND ROBOERTS-CHEBYCHEV THEOREM

Cognate is a linkage of different geometry that generates same coupler curve. One of the unusual properties of the planar four-bar linkage is that there is not one but three four-bar linkages that generate the same coupler curve. This was discovered by Roberts in 1875 and by Chebychev in 1978 and hence is known as the Roberts-chebychev theorem. Though mentioned in an English publication in 1954, it did not appear in the American literature until it was presented independently and almost simultaneously, by Roland T. Hinkle of Michigan state university in 1958. Fig 1.5 shows a four-bar linkage for which other two cognates are to be determined.

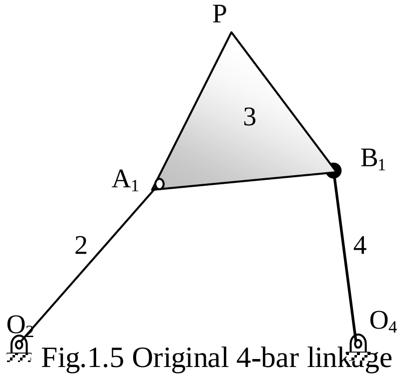


Fig.1.5 Original 4-bar linkage

First step is to release the fixed pivots (O_2 and O_4). While holding the coupler stationary, rotate the links 2 and 4 into co linearity with line of centres (A_1 B_1) of link-3 (See Fig.1.6).

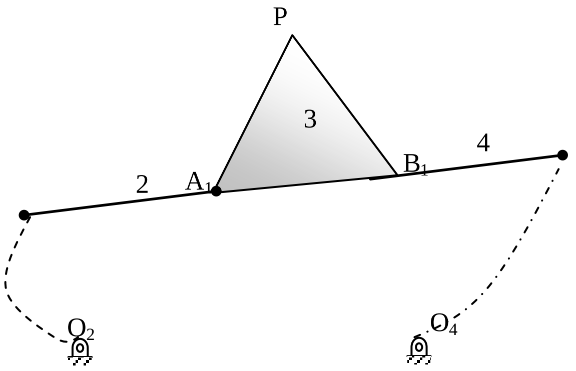


Fig.1.6 Align links 2 and 4 with coupler

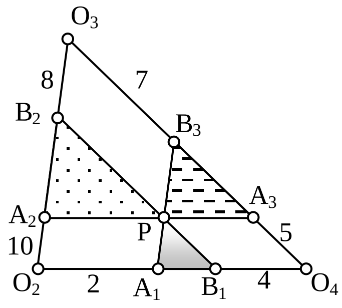


Fig.1.7 Cayley diagram

Now construct the lines parallel to all sides of the links in original linkage to create a Cayley diagram (Fig.1.7). This defines the lengths and shapes of the two cognate linkages. All three four-bar linkages share the original coupler point P and generate the same path motion on their coupler curves. For finding the location of the pivot O_3 from Cayley diagram, the ends of links 2,4 are returned to the original locations of fixed pivot O_1 and O_2. The other links will follow this motion, maintaining the parallelogram relationship between the links and fixed pivot O_3 will be then in its proper location on the ground plane. This configuration is called Roberts diagram and is shown in Fig.1.8.

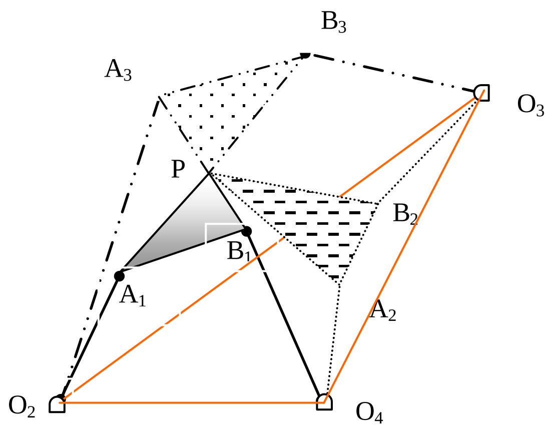


Fig.1.8 Robert's diagram

Robert's diagram can be drawn directly from original linkage without resort to the Cayley diagram by noting that the parallelograms which form the other cognates are also present in Robert diagram.

1.5 LITERATURE REVIEW

Following works relating to the synthesis of four-bar linkage using optimization methods are surveyed briefly.Cabrera *et al.* [1] employed genetic algorithm method to solve the optimum synthesis problem of the four-bar mechanism. Here two constraints namely Grashof's criteria and sequence of the input angles have been considered and constraints are handled by penalty approach. Later on several others purposed different optimization techniques to obtain the optimal solution of the large sized path synthesis problem. Kinzel *et al.*[2] proposed geometric constrained programming approach for kinematic synthesis of planar mechanism. Smaili & Diab[3] adopted ant gradient algorithm for synthesis of four-bar synthesis mechanism and introduced hybrid task mechanism concept. Acharaya & Mandal[4] employed particle swam optimization (PSO) & differential evolution (DE) approach for synthesis of 4-bar linkages. Zadah *et al.*[5] used multi-objective genetic algorithms (GA) for pareto optimal synthesis of 4-bar linkages. Two objectives namely tracking error and transmission angle deviation from 90 degree is accounted. Erkaya and Uzmay [6] presented a joint clearance influence on path generation and transmission angle by adapting GA. Memertas [7] proposed a real-coded evolutionary algorithm for path synthesis of 4-bar linkage, obtained as a combination of DE and real-coded GA methods. Mc Dougall & Nokleby [8] developed & implemented a distributed variant of multi-objective PSO method for 4-bar linkage synthesis. Mermerta [9] presented optimal kinematic design of planar manipulator with four-bar mechanism. As a result of this, it is shown that based on the determined link measurements performance of the manipulator can be maximum not only for a certain position, but also for a position interval. Todorov [10] described a new dimensional synthesis method. The position function of the four-bar mechanism is presented by the Freudenstein's equation and it is minimized by the Chebyshev's best approximation theory. Khare & Dave[11] developed a closed form equations are developed for the synthesis of the 4-bar crank-rocker mechanism in which the angle between dead-centre positions of the rocker and the corresponding angle turned by the crank are prescribed. Ahmed and Waldron [12] outlined synthesis techniques for 4-bar linkages, having adjustable driven crank pivots, for different motion generation problems. The method of solution is analytical in nature, and, therefore well suited for use on a digital computer. Levitskii *et al.*[13] considered the general problem of determining five parameters specifying a for 4-bar linkages which synthesizes a given function and at the same time satisfies some limiting conditions. Hobson & Torfason [14] presented the design of mechanism, which approximate desired centrodes and the applied to two prosthetic knee mechanisms. Kunjur and Krishnamurty [15] employed GA to show multi-point path synthesis problems of 4-bar linkages. Roston and Sturges [16] presented GA approach by relaxing the accuracy of precision points. Geem & Kim [17] presented a new structural optimization method based on the harmony search algorithm. Mahdavi, Fesanghary & Damangir [18] presented an improvised HS algorithm for solving optimization problem.

1.6 OBJECTIVES OF PRESENT WORK

Many techniques for the synthesis of linkages are invented in recent years. Most of these approaches are involved techniques and are mathematically complicated. Only few of them allow a closed from solution. Of these, optimization procedures attempting to minimize an objective function play an important role. A set of inequality constraints that limit the range of variation of parameters may be included in the calculation. The new values of linkage parameters are generated with each iteration step according to particular optimization scheme used. The closest achievable fit between the calculated points and desired points is sought. Even the desired points will not exactly match but this is considered as acceptable result for most engineering tasks. Each optimization approach has as own advantages and disadvantages in term of convergence accuracy, reliability, complexity and speed. Some methods converge even to a minimum value of objective they may not be the best solution. Based on this points there is a lot of scope for application of new methods of optimization for four-bar synthesis problem. Following are the main objective of the present work:

(1) Define the four-bar synthesis problem as a constrained optimization problem.

(2) Implementation of harmony search (HS) optimization scheme.

(3) Validate the results with benchmark examples available in literature with other methods.

The organization of thesis is as follows:

Chapter-2 explains the basis path synthesis problem in terms of objective function to be minimized and constraints to be satisfied.

Chapter-3 describes the history and algorithm of harmony search optimization method adopted in present work

Chapter-4 gives the methodology implemented in MATLAB for handling constraints and minimizing the objective function. Results of four test cases are illustrated to show the effectiveness of method. Chapter-5 presents summary and future scope of the work.

CHAPTER-2

PATH SYNTHESIS OF FOUR-BAR LINKAGE

2.1 COUPLER POINT COORDINATES

In the problem of four-bar linkage synthesis there is some number of precision points to be traced by the coupler point P. To trace the coupler point, the dimension of the links (a, b, c, d, L_x, L_y) is to be determined along with the input crank angle θ_2, so that the average error between these specified precision points (Px_{di}, Py_{di}), (where i=1,2,…N with N as number of precision points given) and the actual points to be traced by the coupler point P gets minimized. The objective or error function can be calculated when the actual traced points (Px_i, Py_i) is evaluated which is traced by the coupler point P with respect to the main coordinate from X,Y as shown in Fig.2.1.

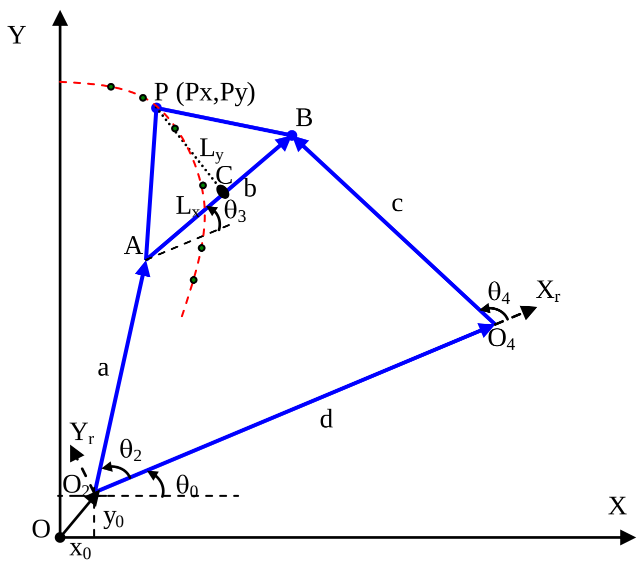


Fig.2.1 Four-bar linkage with ABP as coupler link

The position vector of the coupler point P reference frame X_r-Y_r can be expressed as a vector equation:

$$\vec{r}^{\,P} = \vec{a} + \vec{L}_x + \vec{L}_y \tag{2.1}$$

which can be represented in its components according to:

$$Px_r = a\cos\theta_2 + L_x\cos\theta_3 + L_y(-\sin\theta_3) \tag{2.2}$$

$$Py_r = a\sin\theta_2 + L_x\sin\theta_3 + L_y\cos\theta_3 \tag{2.3}$$

Here, for calculation the coupler point coordinates (Px, Py), we have to first compute the coupler link angle θ_3 using the following vector loop equation for the four-bar linkage:

$$\vec{a}+\vec{b}-\vec{c}-\vec{d}=0 \quad (2.4)$$

This equation also can be expressed in its components with respect to relative coordinates:

$$a\cos\theta_2 + b\cos\theta_3 - c\cos\theta_4 - d = 0 \quad (2.5)$$

$$a\sin\theta_2 + b\sin\theta_3 - c\sin\theta_4 = 0 \quad (2.6)$$

We can compute the angle θ_3 for known values of θ_2 and eliminating θ_4 so, the result will be

$$K_1\cos\theta_3 + K_4\cos\theta_2 + K_5 = \cos(\theta_2-\theta_3) \quad (2.7)$$

where $K_1=d/a$, $K_4=d/b$ and $K_5=\dfrac{c^2-d^2-a^2-b^2}{2ab}$ (2.8)

For this equation following two solutions are obtained:

$$\theta_3^{1} = 2\tan^{-1}\left(\frac{-E+\sqrt{E^2-4DF}}{2D}\right) \quad (2.9)$$

$$\theta_3^{2} = 2\tan^{-1}\left(\frac{-E-\sqrt{E^2-4DF}}{2D}\right) \quad (2.10)$$

where $D=\cos\theta_2 - K_1 + K_4\cos\theta_2 + K_5$, $E=-2\sin\theta_2$ and $F=K_1+(K_4-1)\cos\theta_2+K_5$ (2.11)

These solutions may be (i) real and equal (ii) real and unequal and (iii) complex conjugates. If the discriminant E^2-4DF is negative, then solution is complex conjugate, which simply means that the link lengths chosen are not capable of connection for the chosen value of the input angle θ_2. This can occur either when the link lengths are completely incapable of connection in any position. Except this there are always two values of θ_3 corresponding to any one value of θ_2. These are called, (i) crossed configuration (plus solution) and (ii) Open configuration of the linkage (minus solution) and also known as the two circuits of the linkage. The other methods such as Newton-Raphson solution technique can also be used to get approximate solution for θ_3. The position of coupler P, with respect to world coordinate system XOY is finally defined by:

$$Px = x_0 + Px_r\cos\theta_0 - Py_r\sin\theta_0 \quad (2.12)$$

$$Py = y_0 + Px_r\sin\theta_0 + Py_r\cos\theta_0 \quad (2.13)$$

2.2 POSITION ERROR AS OBJECTIVE FUNCTION

The objective function is usually used to determine the optimal link lengths and the coupler link geometry. In path synthesis problems, this part is the sum squares which computes the position error of the distance between each calculated precision point (Px_i, Py_i) and the desired points (Pxd_i,Pyd_i) which are the target points indicated by the designer. This is written as:

$$f(X) = \sum_{i=1}^{N}[(Pxd_i - Px_i)^2 + (Pyd_i - Py_i)^2] \tag{2.14}$$

where X is set of variables to be obtained by minimizing this function. Some authors have also considered additional objective functions such as the deviation of minimum and maximum transmission angles μ_{min} and μ_{max} from 90^o, for all the set of initial solutions considered.

2.3 THE CONSTRAINTS OF THE LINKAGE

The synthesis of the four-bar mechanism greatly depends upon the choice of the objective function and the equality or the inequality constraints which is imposed on the solution to get the optimal dimensions. Generally the objective function is minimized under certain conditions so that the solution is satisfied by a set of the given constraints. The bounds for variables considered in the analysis are treated as one set of constraints, while the other constraints include: Grashof condition, input link order constraint and the transmission angle constraint.

(1) Grashof criterion

For Grashof criterion, it is required that one of the links of mechanism, should revolve fully by 360^o angle. There are three possible Grashof linkages for a four-bar crank chain: (a) Two crank-rocker mechanisms (adjacent link to shortest is fixed) (b) One double crank mechanism (shortest link is fixed) and (c) One double rocker mechanism (opposite to shortest link is fixed). Of all these, in the present task, only crank-rocker mechanism configuration is considered. Here, the input link of the four-bar mechanism to be crank. Grashof criterion states that the sum (Ls+Ll) of the shortest and the longest links must be lesser than the sum (La+Lb) of the rest two links. That is:

$(Ls+Ll \leq La+Lb)$

(or) $2(Ls+Ll) \leq a+b+c+d$

(or) $g_{1=}\ 2(Ls + Ll)/\ (a + b + c + d)\ - 1 \leq 0$ (2.15)

In the present work violation is defined as follows :

Grashof's = 1 if $g_1 > 0$

Or =0 if $g_1 \leq 0$

(2) Input link angle order Constraint

Usually a large combinations of the mechanisms exists that generates the coupler curves passing through the desired points, but those solutions may not satisfy the desired order. To ensure that the final solution honours the desired order, testing for any order violation is imposed. This is achieved by requiring that the direction of rotation of the crank as defined by the sign of its angular increments $\Delta\theta_2^i=(\theta_2^i-\theta_2^{i-1})$, between the two positions i and i-1, where i=3,4,5,.....,N, have same direction as that between the 1st and the 2nd positions $(\theta_2^2-\theta_2^1)$. That check the following:

Is $\text{sign}(\Delta\theta_2^i)==\text{sign}(\theta_2^2-\theta_2^1)$ for all i=3 to N? (2.16)

where sign(Z)=1 if $Z\geq0$

=-1 if $Z<0$ (2.16a)

If this condition is not satisfied the solution is rejected.

(3) Transmission Angle Constraint

For a crank-rocker mechanism generally the best results the designers recognize when the transmission angle is close to 90 degree as much as possible during entire rotation of the crank. Alternatively, the transmission angle during entire rotation of crank should lie between the minimum and maximum values. This can be written as one of the constraints as follows. First of all, the expressions for maximum and minimum transmission angles for crank-rocker linkage are defined.

$$\mu_{max}=\cos^{-1}\left[\frac{b^2-(d+a)^2+c^2}{2bc}\right]$$

$$\mu_{min}=\cos^{-1}\left[\frac{b^2-(d-a)^2+c^2}{2bc}\right] \quad (2.17)$$

The actual value of transmission angle at any crank angle θ_2^i is given by:

$$\mu=\cos^{-1}\left[\frac{b^2-a^2-d^2+c^2+2ad\cos\theta_2^i}{2bc}\right] \quad (2.18)$$

The condition to be satisfied is: $\mu_{min} \leq \mu \leq \mu_{max}$ (2.19)

The constraint given by equations (2.15), (2.16) and (2.19) are handled by penalty method. That is the non-dimensional constraint deviation is directly added to the objective function for minimization.

For example, constraint eq.(2.19) if not satisfied, the penalty term is given as follows:

$$\text{Trans}=\sum_{i=1}^{N}(1-\text{Trans}\min)(\mu_i-\mu_{\min})^2+(1-\text{Trans}\max)(\mu_i-\mu_{\max})^2$$

where

$\text{Transmin}= \text{sign}(b^2+c^2-(d-a)^2-2bc\cos\mu_{min})$

$\text{Transmax}=\text{sign}(2bc\cos\mu_{max}-b^2+c^2+(a+d)^2)$

Thus the solution seeks to obtain a feasible set of optimum values.

4. Variable Bounds

All variables considered in the design vector should be defined within prespecified minimum and maximum values. Often, this depends on the type of problem. For example, if we have 19 variables in a 10 point optimization problem, all the variables may have different values of minimum and maximum values. Generally, in non-conventional optimization techniques starting with set of initial vectors, this constraint is handled at the beginning itself, while defining the random variable values. That is we use the following simple generation rule:

$$X=X_{min}+\text{rand}\,(X_{max}-X_{min})$$

Where rand is a random number generator between 0 and 1.

2.4 OVERALL OPTIMIZATION PROBLEM

The objective function is the sum of the error function and the penalties assessed to violation the constraints as follows:

$F(k)=f(X) + W1 \times \text{Grashof} + W2\times \text{Tran}$, whereas

W1 is the weighting factor of the Grashof's criteria and W2 is the weighting factor of the Transmission angle constraints .these additional terms acts as scaling factors to fix the order of magnitude of the different variables present in the problem or the objective function..

CHAPTER-3

HARMONY MEMORY SEARCH METHOD

3.1 BASIC ALGORITHM

The HS algorithm conceptualizes a behavioural phenomenon of musicians in the improvisation process, where each musician continues to experiment and improve his or her contribution in order to search for a better state of harmony. It is first given by Geem & Kim [17]. This section describes the HS algorithm based on the heuristic algorithm that searches for a globally optimized solution. The procedure for a harmony search, which consists of steps 1-5.

Step 1. Initialize the optimization problem and algorithm parameters.

Step 2. Initialize the harmony memory (HM) .

Step 3. Improvise a new harmony from the HM .

Step 4. Update the harmony memory.

Step 5. Repeat steps 3 and 4 until the termination criterion is satisfied.

These steps are explained below:

Step 1: Initialize the optimization problem and algorithm parameters. First, the optimization problem is specified as follows:

$$\text{Minimize } f(X) \text{ subjected to } x_i \in X,\ i=1, 2......N \tag{3.1}$$

Where f(X) is an objective function; X is the set of decision variables; N is the number of decision variable; X is the set of the possible range of values for each decision variable, that is, $x_i^L \le x \le x_i^U$ and xi^L and xi^U are the lower and the upper bounds for each decision variables, respectively. The algorithm requires several parameters: Harmony memory size (HMS), Maximum number of improvisations (NI) Harmony Memory Consideration Rate (HMCR), pitch adjusting rate (PAR), Bandwidth vector used in (bm).

Step 2: The HM matrix is initially filled with as many randomly generated solution vectors as the HMS, as well with the corresponding function values of each random vector, f(X). This is shown below:

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}-1} & x_2^{\mathrm{HMS}-1} & \cdots & x_{N-1}^{\mathrm{HMS}-1} & x_N^{\mathrm{HMS}-1} \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_{N-1}^{\mathrm{HMS}} & x_N^{\mathrm{HMS}} \end{bmatrix}.$$

Step-3: A New Harmony vector, $\acute{X}= (x_1', x_2' \ldots, \acute{x}_N)$, is improvised based on the following three mechanisms: (1) random selection, (2) memory consideration, and (3) pitch adjustment. In the random selection, the value of each decision variable, in the New Harmony vector is randomly chosen within the value ranges with a probability of (1-HMCR). The HMCR, which varies between 0 to 1, is the rate of choosing one value from historical values stored in the HM, and (1-HMCR) is the rate randomly selecting one value from the possible range of values.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i{}', \ldots\ldots x_i{}^{HMS}\}\ if\ rand(-1,1) < HMCR, With\ probablitity\ HMCR \\ x_i' \in X_i\ \ othrwise,\quad With\ probablitity\ (1-HMCR) \end{cases} \tag{3.2}$$

The value of each decision variable obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate original pitch obtained in the memory consideration is kept with a probability of HMCR . (1-PAR). If the pitch adjustment decision for x_i' is made with the probability of PAR, x_i' is replaced with $x_i' \pm \mathrm{rand}(-1{,}1)\mathrm{xbw}$, where bw is an arbitrary distance bandwidth for the continuous design variable, and adjustment is applied to each variable as follows:

$$x_i' \leftarrow \begin{cases} x_i' \pm u(-1,1).bw\ With\ probablitity\ PAR(that\ if\ rand(-1,1) < PAR \\ x_i' \qquad\qquad With\ probablitity\ \ (1-PAR) \end{cases} \tag{3.3}$$

Step 4. Update the HM. If the New Harmony vector is better than the worst harmony vector in the HM, based on the evaluation of the objective function value, the New Harmony vector is included in the HM, and the existing worst harmony vector is excluded from the HM.

Step 5. If the stopping criterion (or maximum number of improvisation) is satisfied, the computation is terminated. Otherwise, steps 3 and 4 are repeated.

3.2 IMPROVISED HARMONY MEMORY (HM) ALGORITHM

HMCR, PAR and bw are very important factors for the high efficiency of the HS methods are can be potentially useful in adjusting convergence rate of algorithms to the optimal solutions.

These parameters are introduced to allow the solution to escape from local optima and to improve the global optimum prediction of the HS algorithm. So fine adjustment to these parameters are of great interest Mahdavi et al.[18] introduced bw_i as follows:

$$bw_i = bw_{max} \,.\exp\left[\ln\left(\frac{bw_{min}}{bw_{max}}\right).\frac{i}{NI}\right], \qquad (3.4)$$

Where bw_{min} is the minimum bandwidth and bw_{max} is the maximum bandwidth.

To improve the performance of the HS algorithm and eliminate the drawbacks associated with fixed values of PAR and bw, Mahdavi et al. Proposed an improved harmony search (IHS) algorithm that uses variable PAR and bw in improvisation step. In their method PAR

And bw changes dynamically with generation number as expressed below:

$$PAR(gn) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} \times gn \qquad (3.5)$$

Where the parameters are PAR (gn), pitch adjusting rate for each generation; PAR_{min}, minimum pitch adjusting rate; PAR_{max}, maximum pitch adjusting rate; gn, generation number and bw(gn)= $bw_{max}.\exp(cgn)$.

Whereas, c = $\frac{\ln\left(\frac{bw_{min}}{bw_{max}}\right)}{NI}$.

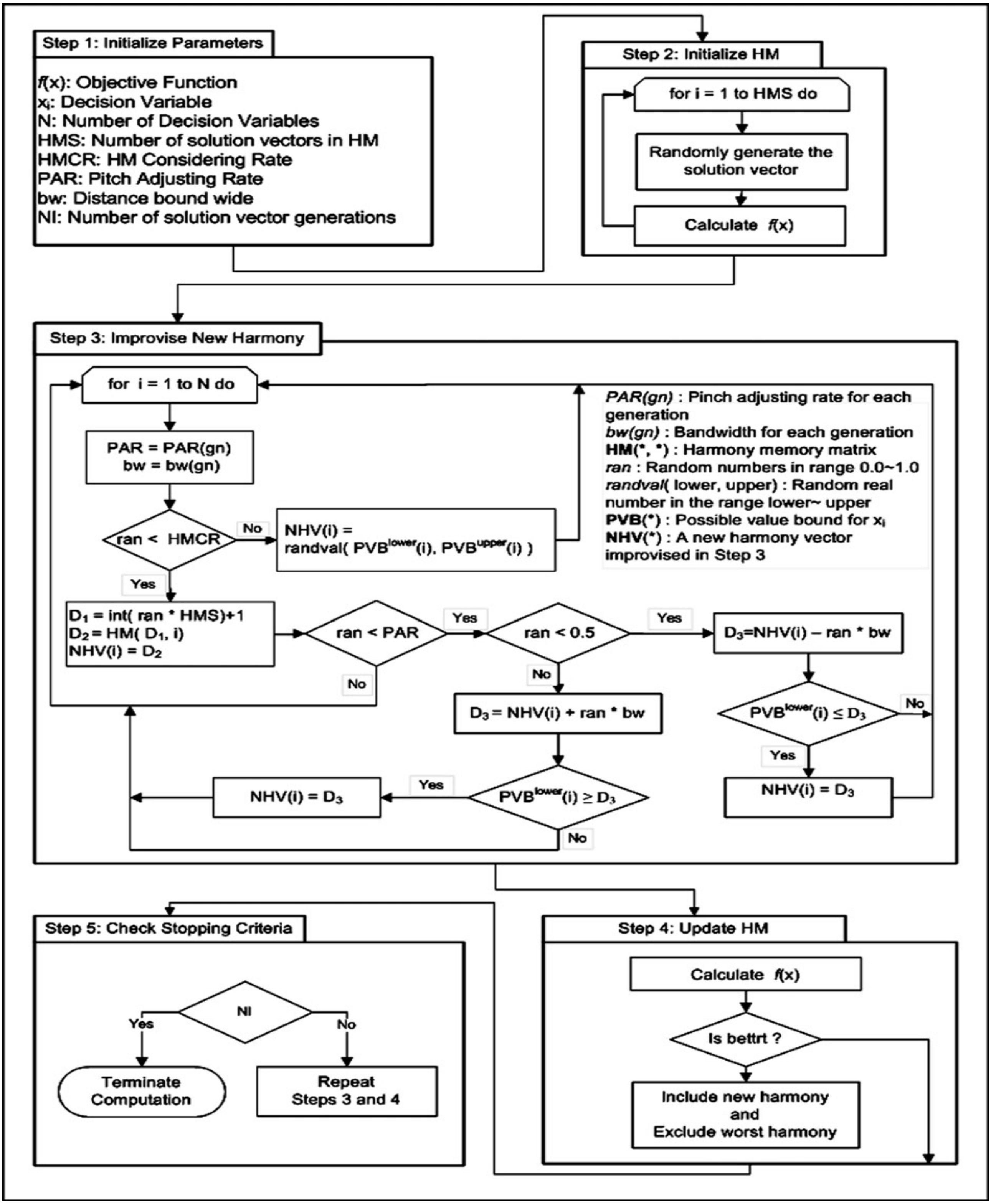


Fig.3.1 Flowchart of IHM algorithm

Where bw (gn) is the bandwidth for each generation; bw_{min} the minimum bandwidth and bw_{max} is the maximum bandwidth .Fig.3.1 shows the flowchart of the improvised HM algorithm adopted in the present work.

3.3 COMPARISONS WITH GENETIC ALGORITHM

Genetic algorithms is widely used method in optimization problems for application of bioinformatics, computational science, engineering, economics, chemistry, manufacturing and other fields. It is based on Darwin's concept of survival of the fittest. Here every generation (cycle) consists of constant number of population size. The objective function to find fittest (maximum) of all in that generation as optimum solution set. It works on crossover and mutation operators. A typical genetic algorithm requires:

1. A genetic representation of the solution domain.
2. A fitness to evaluate the solution domain.

In the genetic algorithm the procedure is stated in steps as below:

1. Choose the initial population of individuals
2. Evaluate the fitness of each individual in that population
3. Repeat on this generation until termination (time limit, sufficient fitness achieved, ect.):
    1) Select the best-fit individuals for production
    2) Breed new individuals through crossover and mutation operation to give birth to offspring
    3) Evaluate the individual fitness of new individuals.

Harmony search algorithm is inspired by the phenomenon of musician attuning .

The harmony memory search algorithm has the following merits:

1. HS does not require different gradients, thus it can consider discontinuous functions as well as continuous function. This is due to the use of stochastic random searches.
2. HS can handle discrete variables as well as continuous variables.
3. HS does not require initial value setting for the variables.

4. HS is free from divergence.
5. HS may escape local optima
6. HS may overcome the drawback of GA's building block well only if the relationship among variables in a chromosome is carefully considered. If neighbour variables in a chromosome have weaker relationship than remote variables, building block theory may not work well because of crossover operation. However, HS explicitly considers the relationship using ensemble operation.
7. HS has a novel stochastic derivation applied to discrete variables, which uses musician's experiences as a searching direction.
8. Certain HS variants do not require algorithm parameters such as HMCR and PAR, thus novice users can easily use the algorithm.

Many similarities are found in HMS compared to GA. Apart from main advantages quoted above; HMS requires less of input data and population size. Also the constraint handling is similar to other approaches, namely one can adopt: penalty method or constraints can be separately handled.

For constrained problems, most applications use penalty function method that transform objective function f(X) into an unconstrained function F(x) consisting of a sum of the objective and the constraints weighted by penalties. By penalty method, the objective is inclined to guide the search towards the feasible solutions. A penalty function method exerts penalty on infeasible solution based on the distance away from the feasible region. The overall objective function is described as :

Minimize F(x) = f(x)+$\lambda\sum_{j=1}^{gn} \max(0, g_j)$, where λ represents penalty coefficient .

CHAPTER-4

RESULTS AND DISCUSSION

4.1 PATH SYNTHESIS WITHOUT PRESCRIBED TIME:

The efficiency and accuracy of the proposed are verified by studying four method cases (for more than five target points) from the literature. Two cases are explained :

(1) 6 points (15 variables)

(2) 10 points (19 variables). Different parameters are used. It includes HS algorithm .

Number of variable NVAR=15, maximum no of iteration Maxitr=10000, harmony memory size HMS=30, harmony memory consideration rate HMCR=0.95, maximum pitch adjustment rate PARmax=0.9, minimum pitch adjustment rate PARmin=0.4, bandwidth minimum bw_{min}=0.0001, bandwidth maximum bw_{max}=1.

(1) Six Points Path Generation and 15 design variables With-out Prescribed Timing :

The first case is a path synthesized problem with given six target points arranged in a vertical line without prescribed timing.

Design variables are :

$$X = [a, b, c, d, l_y, l_x, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_0, y_o, x_o]$$

Target points : $C_d^i = [(20,20), (20,25), (20,30), (20,35), (20,40), (20,45)$

Limits of the variable :

$$a, b, c, d \in [5,60];$$

$$l_x, l_y, x_o, y_o \in [-60,60];$$

$$\theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 \in [0,2\pi]$$

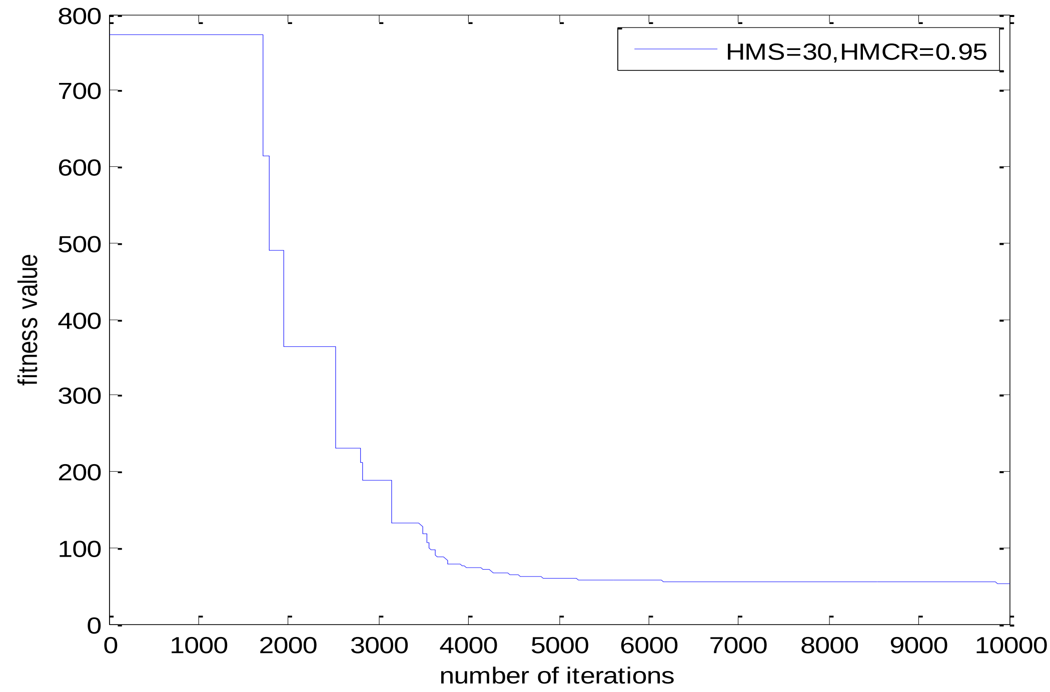
The synthesized geometric parameters and the corresponding values of the precision points (Pxd, Pyd) and the traced points by the coupler point (Px,Py) and the difference between them are shown in table 1 and table 2 respectively . Although the constraint of the sequence of the input angles during the evolution is ignored in this case .The accuracy of the solution in case 1 has been remarkably improved using the present method. Fig(4.1) shows the convergence graph of HS algorithm .Fig(4.2) shows the six target points and the coupler curve obtained using the harmony memory search method with NVAR=15, Maxitr=10000, HMS=30, HMCR=0.95, PARmax=0.9, PARmin=0.4, bw_{min}=0.0001, bw_{max}=1. The time was taken to run the programme in MATLAB was 3.94 seconds to show the out-puts as below.

Table 1. synthesized results for case 1(1).

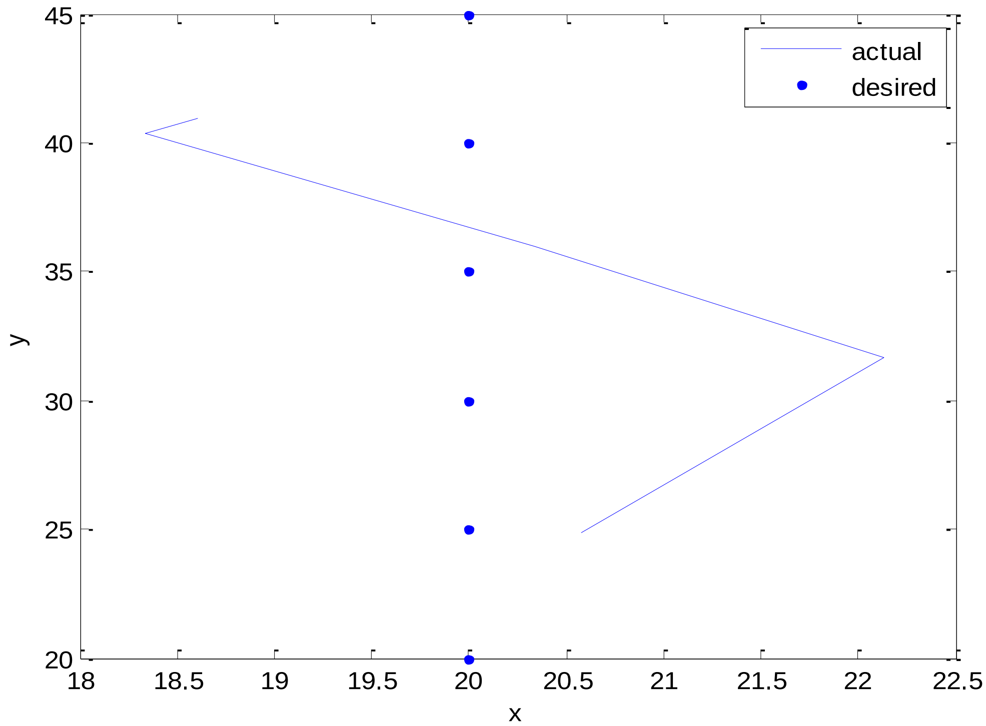
a	b	C	d	lx	ly	Θ1	Θ2	Θ3	Θ4	Θ5	Θ6	θ_o	lx_0	ly_o
12.0718	21.6454	43.9734	53.2298	21.2555	9.2386	1.1338	1.1342	4.4424	4.7418	5.1811	5.9254	2.8427	36.8209	12.1021

Table 2. The actual points which is traced by the coupler link and the precision points.

SL NO	Px	pxd	(Px-pxd)	(px-pxd)²	py	pyd	(py-pyd)	(py-pyd)²
1	20.574815	20.000000	0.574815	0.3304	24.852480	20.000000	4.85248	23.5465
2	20.574583	20.000000	0.574583	0.3301	24.847631	25.000000	-0.152369	0.0232
3	22.133817	20.000000	2.133817	4.5531	24.847631	30.000000	-5.152369	26.5469
4	20.340553	20.000000	0.340553	0.1159	35.957300	35.000000	0.957300	0.9164
5	18.336832	20.000000	-1.663168	2.7661	40.332799	40.000000	0.332799	0.1107
6	18.607485	20.000000	-1.392515	1.9390	40.972467	45.000000	-4.027533	16.2210



Fig(4.1) (variation between fitness and the iteration number)



Fig(4.2) (six target points and the coupler curve obtained).

(2) Ten Points Path Generation and 19 design variables With-out Prescribed Timing :

Design variables are :

$$X = [a, b, c, d, l_y, l_x, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_2^7, \theta_2^8, \theta_2^9, \theta_2^{610}, \theta_0, y_o, x_o]$$

Target points : $C_d^i =$

[(20,10),(17.66,15.142),(11.736,17.878),(5,16.928),(0.60307,12.736),(0.60307,7.2638),

(5,3.0718),(11.736,2.1215),(17.66,4.8577),(20,10)]

Limits of the variable :

$$a, b, c, d \in [5,80];$$

$$l_x, l_y, x_o, y_o \in [-80,80];$$

$$\theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \ldots \theta_2^{10}, \theta_0 \in [2,2pi]$$

The synthesized geometric parameters and the corresponding values of the precision points (Pxd, Pyd) and the traced points by the coupler point (Px,Py) and the difference between them are shown in table 3 and table 4 respectively . Although the constraint of the sequence of the input angles during the evolution is ignored in this case .The accuracy of the solution in case 1 has been remarkably improved using the present method.fig(4.4) shows the ten target points and the coupler curve obtained using the harmony memory search method with NVAR=18, Maxitr=10000, HMS=30, HMCR=0.95, PARmax=0.9, PARmin=0.4, bw_{min}=0.0001, bw_{max}=1. The time was taken to run the programme in MATLAB was 10.09 seconds to show the out-puts as below.

Table 3 Optimized values for the ten target points problem

a	B	C	d	l_x	l_y	θ_2^1	θ_2^2	θ_2^3	θ_2^4
59.1622	62.2648	53.5249	5.0000	27.9595	47.8058	2.4875	2.9001	3.4924	4.3196

θ_2^5	θ_2^6	θ_2^7	θ_2^8	θ_2^9	θ_2^{10}	θ_o	x_o	y_o
5.1309	0.1616	0.8517	1.5584	2.0857	2.4875	4.7531	6.7328	12.5555

Table 4. Actual points which is traced by the coupler link and the precision points

SL NO	Px	Pxd	(Px-pxd)	(px-pxd)²	py	Pyd	(py-pyd)	(py-pyd)²
1	18.967813	20.000000	-1.032187	1.06541	10.030257	10.000000	0.03025	$9.15\text{x}10^{-4}$
2	17.678804	17.660000	0.678804	0.46077	15.153438	15.142000	0.01143	$1.30\text{x}10^{-4}$
3	12.161751	11.736000	0.425751	0.18126	19.461353	17.878000	1.583353	2.50700
4	4.390400	5.000000	-0.6096	0.37161	17.533260	16.928000	0.60526	0.36633
5	2.401224	0.603070	1.79815	3.23335	12.700758	12.736000	-0.03542	$1.241\text{x}10^{-3}$
6	1.554730	0.603070	0.95166	0.90565	7.286766	7.263800	0.02296	$5.27\text{x}10^{-4}$
7	4.283504	5.000000	-0.71649	0.51336	2.143739	3.071800	-0.92806	0.86129
8	12.126017	11.736000	0.39001	0.15211	0.865529	2.121500	-1.2559	1.5774
9	17.445669	17.660000	-0.21433	0.0459	4.993499	4.857700	0.13579	0.01844
10	18.967813	20.000000	-1.0321	1.06541	10.030272	10.000000	0.03027	$9.1\text{x}10^{-4}$

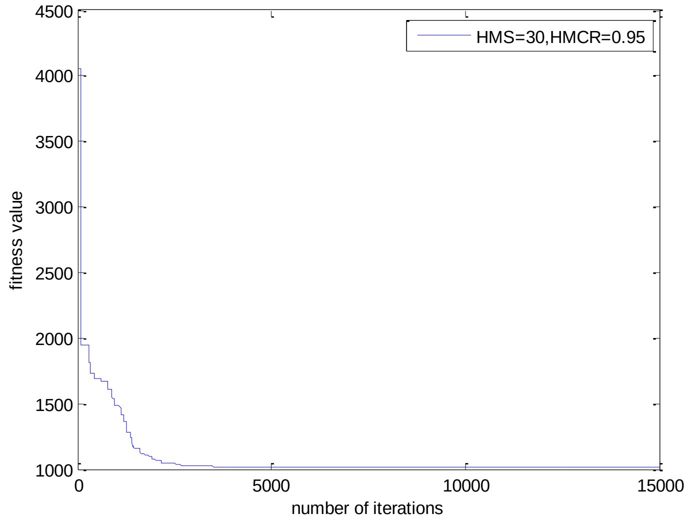


Fig 4.3 (variation between fitness and the iteration number)

Fig 4.3 shows the variation of fitness value with the number of iteration .

Fig 4.4 shows the ten target points and the coupler curve obtained using the harmony memory search method .

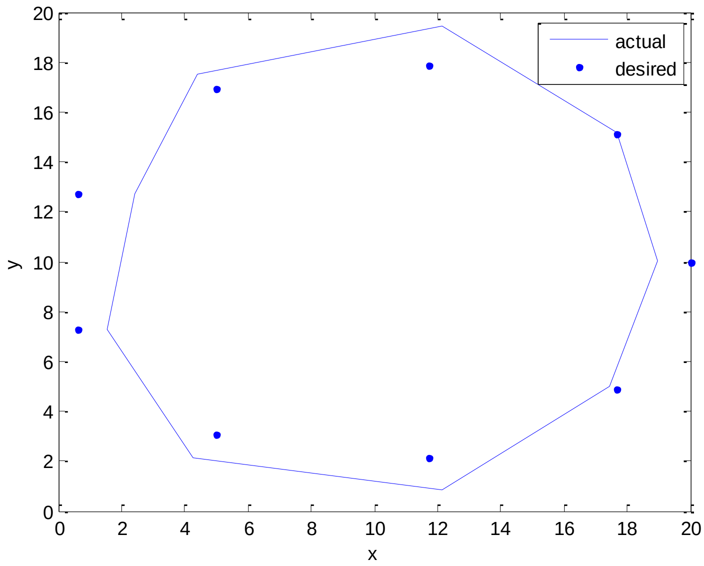


Fig 4.4 (The ten target points and the coupler curve obtained).

4.2 PATH SYNTHESIS WITH PRESCRIBED TIME

(1) Six Points Path Generation and 9 design variables With Prescribed Timing :

Design variables are :

$$X = [a, b, c, d, l_y, l_x, \theta_0, y_o, x_o]$$

Target points : $C_d^i =$

[(0,0),(1.9098,5.8779),(6.9098,9.5106),(13.09,9.5106),(18.09,5.8779),(20,0)]

$$\theta_2^i = [\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, 2 * \frac{\pi}{3}, 5 * \frac{\pi}{6}, \pi]$$

Limits of the variable :

$$a, b, c, d \in [5,50];$$

$$l_x, l_y, x_o, y_o \in [-50,50];$$

$$\theta_0 \in [0,2\pi]$$

The synthesized geometric parameters and the corresponding values of the precision points (Pxd, Pyd) and the traced points by the coupler point (Px, Py) and the difference between them are shown in table 5 and table 6 respectively . Although the constraint of the sequence of the input angles during the evolution is ignored in this case . Following are parameters of HS algorithm NVAR=9, Maxitr=10000, HMS=30, HMCR=0.95, PARmax=0.9, PARmin=0.4, bw_{min}=0.0001, bw_{max}=1. The time was taken to run the programme in MATLAB was 10.97 seconds to show the out-puts as below.

Table 5. synthesized results for case 2(1).

a	b	C	d	l_x	l_y	θ_o	x_o	y_o
13.1108	33.1908	35.8110	10.4905	4.1298	-9.2071	2.6731	18.1286	9.2126

Table 6 . Actual points which is traced by the coupler link and the precision points

SL NO	Px	Pxd	(Px-pxd)	(px-pxd)^2	py	pyd	(py-pyd)	(py-pyd)^2
1	3.631704	0.000000	3.63170	13.1892	18.482750	0.000000	18.4827	341.58
2	3.859645	1.909800	1.94984	3.8018	11.580054	5.877900	5.7021	32.5145
3	7.045476	6.909800	0.13567	0.01840	6.184301	9.510600	- 303262	11.0642
4	11.899930	13.090000	-101907	1.4177	3.436739	9.510600	-6.0738	36.8917
5	16.773960	18.090000	-1.3160	1.7319	3.634789	5.877900	-2.2431	5.0315
6	20.155384	20.000000	0.15538	0.02414	6.171755	0.000000	6.17175	38.088

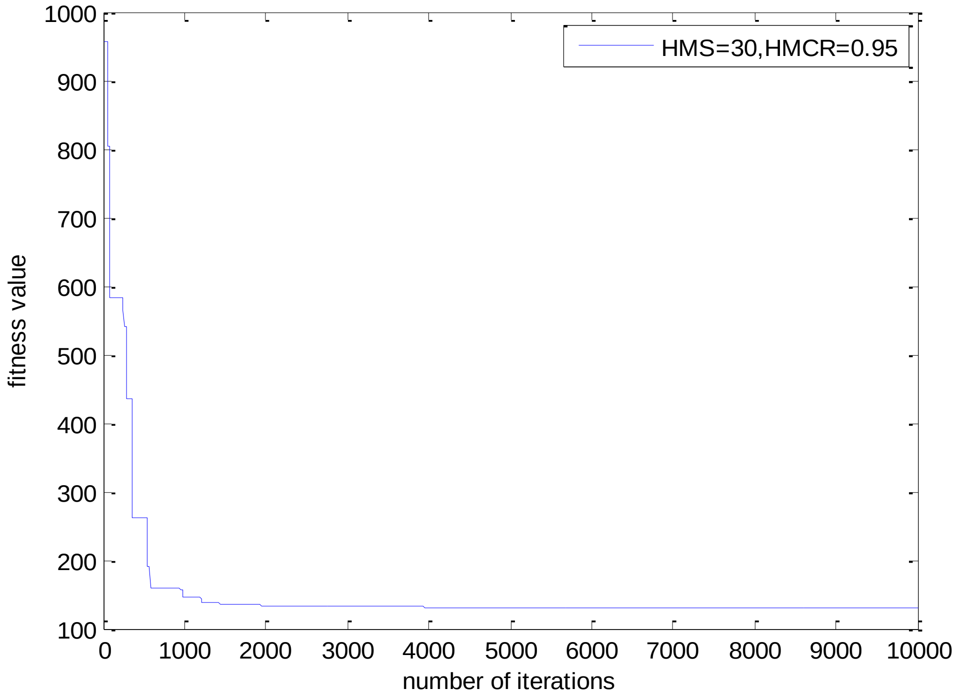


Fig 4.5 (variation between fitness and the iteration number for case 4.2 (1) case)

Fig 4.5 shows the variation of fitness value with the number of iteration .

Fig 4.6 shows the ten target points and the coupler curve obtained using the harmony memory search method.

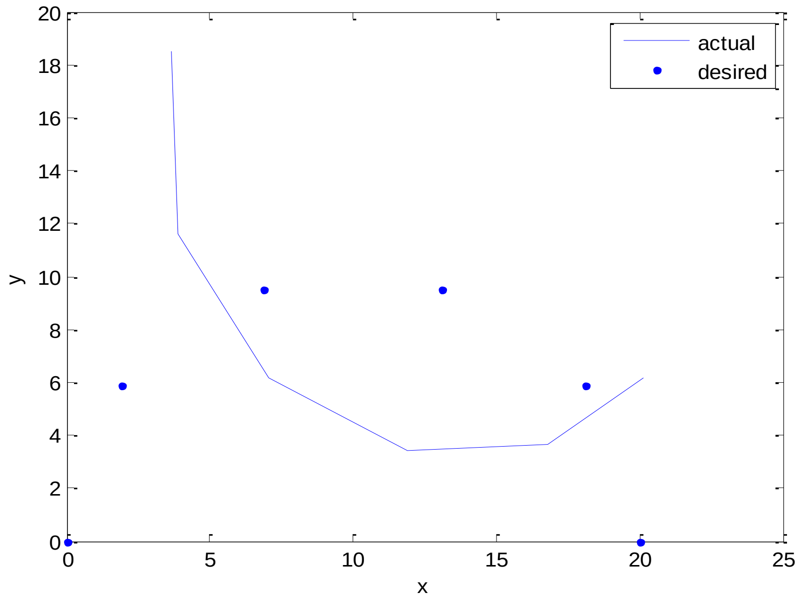


Fig 4.6 (The ten target points and the coupler curve obtained for 4.2(1) case).

(2) Eighteen Points Path Generation and 10 design variables With Prescribed Timing :

Design variables are :

$$X = [a, b, c, d, l_y, l_x, \theta_2^1, \theta_0, y_o, x_o]$$

Target points : $C_d^i =$

[(0.5,1.1),(0.4,1.1),(0.3,1.1),(0.2,1.0),(0.1,0.9),(0.005,0.75),(0.02,0.6),(0,0.5),(0,0.4),(0.03,0.3),(0.1,0.25),(0.15,0.2),(0.2,0.3),(0.3,0.4),(0.4,0.5),(0.5,0.7),(0.6,0.9),(0.6,1)]

$$\theta_2^i = [\theta_2^i . \theta_2^1 + 20 * i\ ](i = 1,2,3..17)$$

Limits of the variable :

$$a, b, c, d\ \in [0,50];$$

$$l_x, l_y, x_o, y_o \in [-50,50];$$

$$\theta_0, \theta_2^i \in [0,2\pi]$$

The synthesized geometric parameters and the corresponding values of the precision points (Pxd, Pyd) and the traced points by the coupler point (Px, Py) and the difference between them are shown in table 7 and table 8 respectively . Although the constraint of the sequence of the input angles during the evolution is ignored in this case . Following are parameters of HS algorithm NVAR=10, Maxitr=10000, HMS=30, HMCR=0.95, PARmax=0.9, PARmin=0.4, bw_{min}=0.0001, bw_{max}=1. The time was taken to run the programme in MATLAB was 04.17 seconds to show the out-puts as below.

Table 7. synthesized results for case 2(2).

a	b	c	d	l_x	l_{yx}	θ_2^1	θ_o	x_o	y_o
0.3526	40.5743	35.8038	35.3361	0.2386	7.1266	1.1442	6.1263	-6.3093	-2.1562

Table 8 . Actual points which is traced by the coupler link and the precision points case 2(2)

SL NO	Px	Pxd	(Px-pxd)	(px-pxd)²	py	Pyd	(py-pyd)	(py-pyd)²
1	0.462358	0.500000	-0.03764	$1.4 * 10^{-3}$	0.891675	1.100000	-0.2083	0.0433
2	0.348859	0.400000	-0.05114	$2.6* 10^{-3}$	0.943100	1.100000	-0.1569	0.0246
3	0.222063	0.300000	-0.07793	$6.0* 10^{-3}$	0.961302	1.100000	-0.1386	0.0192
4	0.097261	0.200000	-0.10273	0.01055	0.943994	1.000000	-0.0560	$3.17.4* 10^{-3}$
5	-0.010518	0.100000	-0.1105	0.01221	0.893198	0.900000	$-6.87.4* 10^{-3}$	$4.7* 10^{-7}$
6	-0.088306	0.005000	-0.09330	$8.7* 10^{-3}$	0.815029	0.750000	0.0650	$4.2* 10^{-3}$
7	-0.126748	0.020000	-0.14674	0.02153	0.718962	0.600000	0.1189	0.0141
8	-0.121214	0.000000	-0.121214	0.01469	0.616669	0.500000	0.1166	0.0136
9	-0.072361	0.000000	-0.072361	$5.2* 10^{-3}$	0.520570	0.400000	0.1205	0.0145
10	0.013947	0.030000	-0.01605	$2.5* 10^{-4}$	0.442301	0.300000	0.1423	0.0202
11	0.127332	0.100000	0.02733	$7.4* 10^{-4}$	0.391287	0.250000	0.1412	0.0199
12	0.254139	0.150000	0.10413	0.01084	0.373610	0.200000	0.1736	0.0301
13	0.379072	0.200000	0.17907	0.03206	0.391312	0.300000	0.0913	$8.3* 10^{-3}$
14	0.487041	0.300000	0.18704	0.03498	0.442187	0.400000	0.0421	$1.2* 10^{-3}$

15	0.564989	0.400000	0.16498	0.02722	0.520083	0.500000	0.0200	$4.0* 10^{-4}$
16	0.603485	0.500000	0.10348	0.01070	0.615653	0.700000	-0.0843	$7.1* 10^{-3}$
17	0.597876	0.600000	$-2.1* 10^{-4}$	$4.7* 10^{-6}$	0.717457	0.900000	-0.1825	0.0333
18	0.548852	0.600000	-0.05114	$2.67.4* 10^{-3}$	0.813303	1.000000	-0.1866	0.0348

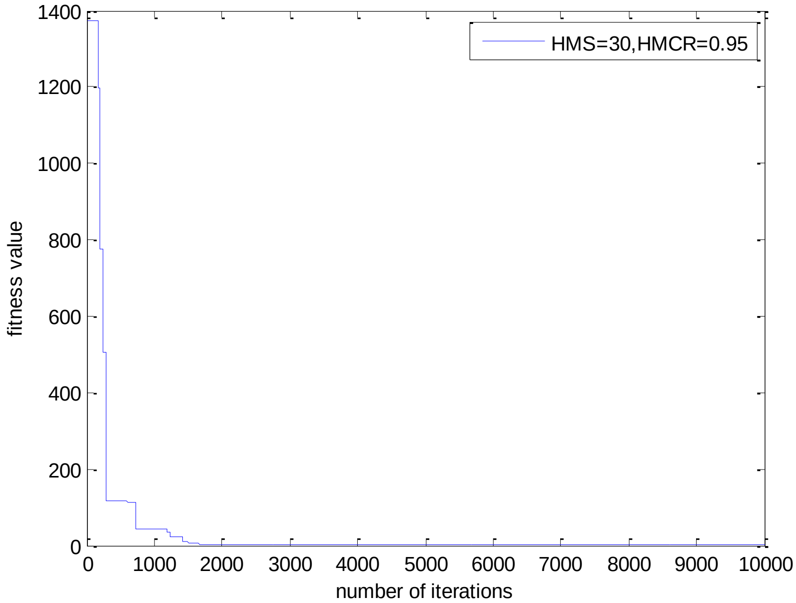


Fig 4.7 (variation between fitness and the iteration number for case 4.2 (1) case)

Fig 4.7 shows the variation of fitness value with the number of iteration .

Fig 4.8 shows the ten target points and the coupler curve obtained using the harmony memory search method.

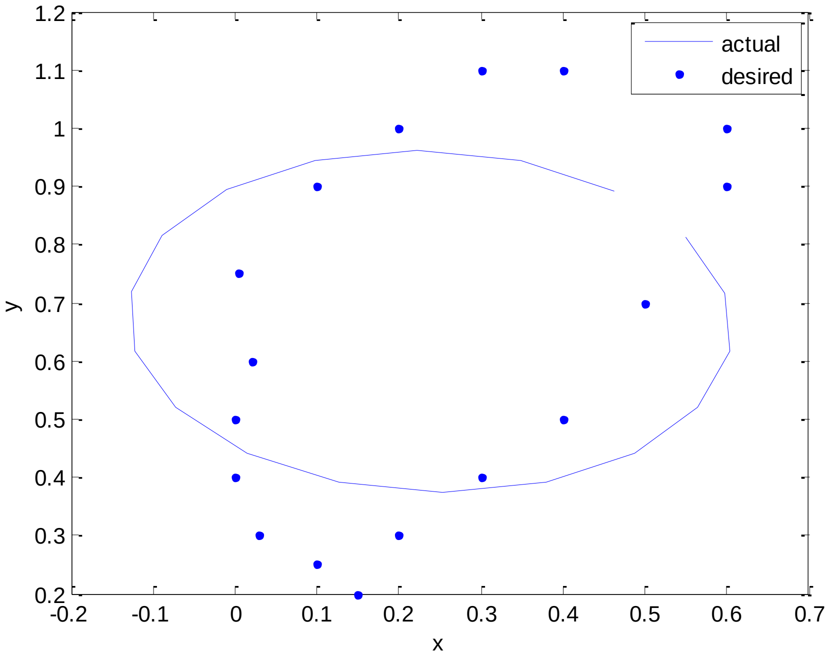


Fig 4.6 (The 18 target points and the coupler curve obtained for 4.2(2) case).

CHAPTER-5

CONCLUSIONS

5.1 SUMMARY OF THE WORK

In the present work we have consider a crank rocker mechanism of four-bar linkage. The objective function namely path error varies with respect to the number of precision points specified. The four different cases with & with-out prescribed timings were considered. It is found that in each case the computational time for convergence of 10,000 cycles changes. In some examples even the constraint violation is maintained, the minimum value of the objective function is found to be close to the published results available in the literature by other methods. In each case the convergence graph, coupler curves & tables of optimum dimensions and final coupler point coordinate were reported.

5.2 FUTURE SCOPE :

Even this work has concentrated on path synthesis part with some important constraints, some more constraints like mechanical advantage of the linkage, and flexibility effects can be also considered to get the accuracy. Also as in hybrid synthesis approach, the same linkage may be adopted both for path synthesis applications as well as motion synthesis applications. The objective function should be modified so as to get a different optimum link dimensions. Finally fabrication of the proto-type of this linkage may be done to know the difference between theoretically obtained coupler coordinates and actual values achieved .

REFRENCES

1) J.A .Carbrea. A. Simon & M.Prado, Optimal synthesis of mechanism with GA, Mechanism & Machine Theory, Vol.37, pp 1165-1177, 2002.

2) E.C.Kinzel,J.P.Schniedeler & G.R.Pennock, Kinetic synthesis for finitely separated positive any geometric coordinate programming, J. Machine design, Trans.ASME, Vol.128,pp.1070-1079,2006

3) A.Smaili and N.Diab, Optimum synthesis of hybrid-task mechanisms using ant-gradient search method, Mechanism & Machine theory, Vol. 42, pp.115-130,2007

4) S.K.Acharyya and M.Mandal, Performance of EAs for four-bar linkage synthesis, Mechanism & Machine Theory, vol. 44, pp.1784-1794,2009

5) N.N.Zadah, M.Fklezi,A.Jamalize M.Garj, Pareto optimal synthesis of four-bar mechanism for path generation, Mechanism & Machine Theory, Vol.44, pp.180-191, 2009.

6) S.Erkaya & I.Uzmay, Determining parameters using GA in mechanism with joint clearance, Mechanism & Machine Theory, Vol.44, pp.222-234, 2009.

7) W.Y.Lin, A GA-DE hybrid evolutionary algorithm for path synthesis of 4-bar linkage', Mechanism & Machine theory, Vol.45, pp.1096-1107, 2010.

8) R.McDougall & S.Nokleby, Grashof's mechanism synthesis using multi-objective parallel asynchronous PSO, Proc- Candian Soc. for mechanical Engg. Forum, Victoria , Jun 7-9,2010,pp.1-7.

9) V.Mermerta, Optimal design of manipulator with four-bars, Mechanism and Machine Theory, Volume 39, Pages 545-554,2004.

10) T.S.Todorov, Synthesis of four-bar mechanisms by Freudenstein–Chebyshev theory, Mechanism and Machine Theory, Volume 37,Pages 1505-1512,2002.

11) A.K. Khare and R. K. Dave, Optimizing 4-bar crank-rocker mechanism, Mechanism and Machine Theory, Volume 14, Pages 319-325,1979.

12) A.Ahmad and K.J.Waldron, Synthesis of adjustable planar 4bar mechanisms, Mechanism and Machine Theory, Volume 14, pp.405-411,1979 .

13) N.I.Levitskii, Y.L.Sarkissyan and G.S.Gekchian, Optimum synthesis of for 4-bar function generating, Mechanism and Machine Theory, Volume 7, Pages 387-398,1972.

14) D.A. Hobson and L.E. Torfason, optimization of four-bar knee mechanisms A computerized approach, Journal of Biomechanics, Volume 7, Pages 371-376,1974.

15) A.Kunjur and S.Krishnamurty, GA in mechanical synthesis, Applied mechanisms and robotics, Vol.4, pp.18-24, 1997.

16) G.P.Roston and R.H.Sturges, GA synthesis of 4 bar linkages, AI for Engg.Design, Analysis and Manufacturing, Vol.10, pp.371-390, 1996.

17) Z.W.Geem & J.H.kim, 'A new heurisrc optimization algorithm: HS', Simulation, vol.76, pp.60-68, 2001.

18) M.Mahdavi, M.Fesanghary & E.Damangir, An improved HS algorithm for solving optimization problem, Applied Mathematical computation, Vol.188, pp-1567-1579, 2007.

19) Robert L.Norton, Design of machinery, Mc Graw hill.

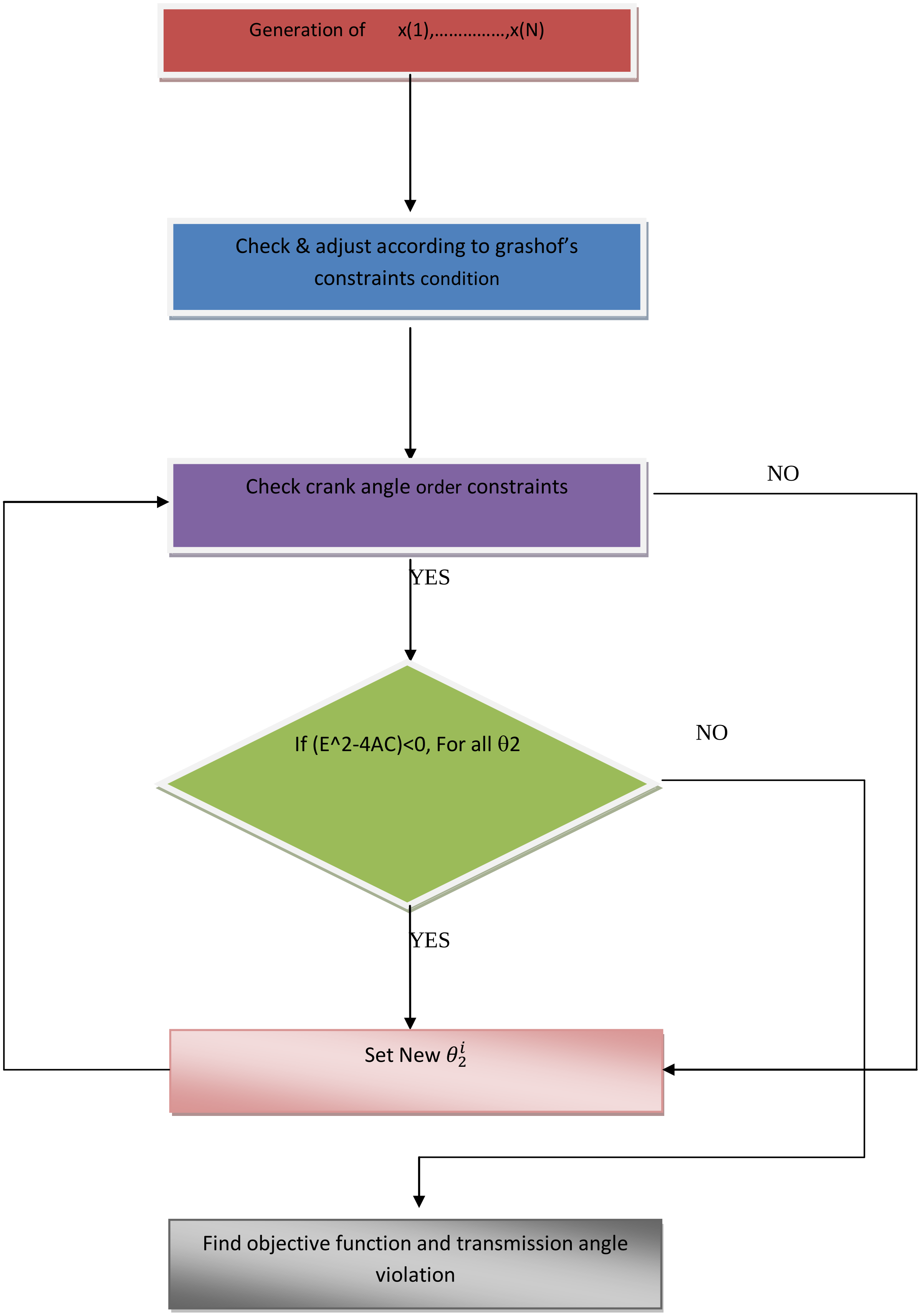
20) John J.Uicker, ect, theory of mechanics and mechanisms, oxford university press.

21) http://en.wikipedia.org/wiki/Harmony_search.

22) http://en.wikipedia.org/wiki/Genetic_algorithm.

APPENDIX

Matlab programs for four cases considered in the work are based on the following flowchart :



A program is written in Matlab for the first problem which consists of three files namely as hmsc1, feasble1 and pathec1 made for path synthesis without prescribed time.

For six points as below :

First file : hmsc1 (Main file)

```
% SIX POINTS WITHOUT PRESCRIBED TIMING-CASE-1

clc
clear;
PVB=[5 60;5 60;5 60;5 60;-60 60;-60 60;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0
2*pi;-60 60;-60 60];
pxd=[20 20 20 20 20 20];
pyd=[20 25 30 35 40 45];

NVAR=15;
Maxitr=10000;
HMS=30;
HMCR=0.95;
PARmax=0.9;PARmin=0.4;
bwmin=0.0001;bwmax=1;

HM=zeros(HMS,NVAR);NCHV=zeros(1,NVAR);
for i=1:HMS
   for j=1:NVAR
      HM(i,j)=PVB(j,1)+rand*(-PVB(j,1)+PVB(j,2));
   end
   X=HM(i,:);
   Y=feasble1(X);
% EVALUATION OF OBJECTIVE AND CONSTRAINT-VIOLATION VALUES
[obj,const]=pathec1(Y);
fitness(i)=obj+const;
HM(i,:)=Y;
 end
 for ite=1:Maxitr
   PAR=(PARmax-PARmin)/(Maxitr)*(ite-1)+PARmin;
   coef=log(bwmin/bwmax)/Maxitr;
   BW=bwmax*exp(coef*ite);
   for i=1:NVAR
      ran=rand(1);
      if(ran<HMCR) % MEMORY CONSIDERATION
         index=floor(1+rand*(-1+HMS));
         NCHV(i)=HM(index,i);
         pvbran=rand(1);
         if(pvbran<PAR)% PITCH ADJUSTMENT
            pvbran1=rand(1);
            result=NCHV(i);
            if(pvbran1<0.5)
               result=result+rand(1)*BW;
               if(result<PVB(i,2))
                  NCHV(i)=result;
               end
            else
               result=result-rand(1)*BW;
               if(result>PVB(i,1))
                  NCHV(i)=result;
               end
            end
         end
      else
         NCHV(i)=PVB(i,1)+rand*(-PVB(i,1)+PVB(i,2));% RANDOM SELECTION
```

```
      end
   end
   X=NCHV;
   Y=X;%feasble1(X);

    [obj,const]=pathec1(Y);
    newfitness=obj+const;

    NCHV=Y;
bestfit=fitness(1);bestindex=1;
for i=2:HMS
   if(fitness(i)<bestfit)
      bestfit=fitness(i);
      bestindex=i;
   end
end
worstfit=fitness(1);worstindex=1;
for i=2:HMS
   if(fitness(i)>worstfit)
      worstfit=fitness(i);
      worstindex=i;
   end
end
if(newfitness<worstfit)
   if(newfitness<bestfit)
      HM(worstindex,:)=NCHV;
      bestgen=NCHV;
      fitness(worstindex)=newfitness;
      bestindex=worstindex;
   else
      HM(worstindex,:)=NCHV;
      fitness(worstindex)=newfitness;
   end
   worstfit=fitness(1);worstindex=1;
   for i=2:HMS
      if(fitness(i)>worstfit)
         worstfit=fitness(i);
         worstindex=i;
      end
   end
end
bestfitness=min(fitness);
fu(ite)=bestfitness;
end
fprintf('optimized values are:\n');
disp(HM(bestindex,:));
for i=1:NVAR
   X(i)=HM(bestindex,i);
end
figure(1);
I=1:Maxitr;
plot(I,fu);

xlabel('number of iterations');
ylabel('fitness value');
legend('HMS=30,HMCR=0.95');
for i=1:HMS
        for j=1:NVAR
           fprintf('%3.2f\t',HM(i,j));
        end
        fprintf('%f\t',fu(Maxitr));
        fprintf('\n');
     end

  k1=X(4)/X(1);k2=X(4)/X(3);k3=(X(1)^2+X(3)^2+X(4)^2-X(2)^2)/(2*X(1)*X(3));
  k4=X(4)/X(2);k5=(-X(1)^2+X(3)^2-X(4)^2-X(2)^2)/(2*X(1)*X(2));
  R=[cos(X(13)) -sin(X(13));sin(X(13)) cos(X(13))];
  T=[X(14);X(15)];
```

```
j2=6;
 for j2=6:11
    E=-2*sin(X(j2+1));
    D=cos(X(j2+1))-k1+k4*cos(X(j2+1))+k5;
    F=k1+(k4-1)*cos(X(j2+1))+k5;i1=j2+1;
    theta_31=2*atan((-E+sqrt(E^2-4*D*F))/(2*D));
    px1=X(1)*cos(X(i1))+X(5)*cos(theta_31)-X(6)*sin(theta_31);
    py1=X(1)*sin(X(i1))+X(5)*sin(theta_31)+X(6)*cos(theta_31);
    P1=[px1;py1];
    P01=R*P1+T;
    px01=P01(1);py01=P01(2);
    xa(j2-5)=px01;
    ya(j2-5)=py01;
 end
 for i=1:6
    fprintf('%f\t%f\t',xa(i),pxd(i));
    fprintf('%f\t%f\n',ya(i),pyd(i));
 end
 figure(2);


 plot(xa,ya,pxd,pyd,'b.');

 xlabel('x');ylabel('y');legend('actual','desired');
```

second file : feasible1 (Function file making the initial solution feasible)

```
function Y=feasble1(X)
VB=[5 60;5 60;5 60;5 60;-60 60;-60 60;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0 2*pi;0
2*pi;-60 60;-60 60];
for j3=1:15
   Xlow(j3)=VB(j3,1);
   Xhigh(j3)=VB(j3,2);
end

% VERIFICATION OF GRASHOF CRITERION IN INITIAL SET
            Llink=max([X(1) X(2) X(3) X(4)]);
            Slink=min([X(1) X(2) X(3) X(4)]);
        while 2*(Llink+Slink)>X(1)+X(2)+X(3)+X(4)
              for j=1:4
                 X(j)=Xlow(j)+rand*(-Xlow(j)+Xhigh(j));
            end
            Llink=max([X(1) X(2) X(3) X(4)]);
        Slink=min([X(1) X(2) X(3) X(4)]);
         end
          j1=7;
            while j1<12
                if (X(j1+1)-X(j1))<0
                    X(j1+1)=Xlow(j1+1)+rand*(-Xlow(j1+1)+Xhigh(j1+1));
                    j1=j1;
                else
                j1=j1+1;
                end
             end
     j2=6;
     while j2<12
        k1=X(4)/X(1);k2=X(4)/X(3);k3=(X(1)^2+X(3)^2+X(4)^2-X(2)^2)/(2*X(1)*X(3));
            k4=X(4)/X(2);k5=(-X(1)^2+X(3)^2-X(4)^2-X(2)^2)/(2*X(1)*X(2));
        E=-2*sin(X(j2+1));D=cos(X(j2+1))-k1+k4*cos(X(j2+1))+k5;
        F=k1+(k4-1)*cos(X(j2+1))+k5;
% CHECK FOR E^2-4*D*F>0 TO PROCEED FURTHER
        if E^2-4*D*F<0
          for j3=1:12
             X(j3)=Xlow(j3)+rand*(Xhigh(j3)-Xlow(j3));
```

```
          end
          Llink=max([X(1) X(2) X(3) X(4)]);
            Slink=min([X(1) X(2) X(3) X(4)]);
        while 2*(Llink+Slink)>X(1)+X(2)+X(3)+X(4)
              for j=1:4
                 X(j)=Xlow(j)+rand*(-Xlow(j)+Xhigh(j));
            end
            Llink=max([X(1) X(2) X(3) X(4)]);
        Slink=min([X(1) X(2) X(3) X(4)]);
         end

          % VERIFICATION OF INPUT LINK ANGLE ORDER CONSTRAINT
            j1=7;
            while j1<12
                if (X(j1+1)-X(j1))<0
                    X(j1+1)=Xlow(j1+1)+rand*(-Xlow(j1+1)+Xhigh(j1+1));
                    j1=j1;
                else
                j1=j1+1;
                end
            end
           j2=j2;
       else
          j2=j2+1;
       end
    end
    Y=X;
```

Third file :pathec1 (Function file to estimate path error and constrain violation)

```
function [obj,const]=pathec1(X)
% objective square error
% DESIRED X AND Y POSITIONS (6 POINT PROBLEM)
pxd=[20 20 20 20 20 20];
pyd=[20 25 30 35 40 45];
error=0;

% Grashof's criteria
LS=min([X(1),X(2),X(3),X(4)]);
LL=max([X(1),X(2),X(3),X(4)]);
if (2*(LS+LL)>X(1)+X(2)+X(3)+X(4))
    CV1=1;%(2*(LS+LL))/(X(1)+X(2)+X(3)+X(4))-1;
else
   CV1=0;
end
% angle sequence constraint
CV2=0;
j1=7;
while j1<12
    if (X(j1+1)-X(j1))<0
       CV2=1;
       break;
    else
    j1=j1+1;
   end
  end
% TRANSMISSION ANGLE constraints (MU_MIN<MU<MU_MAX)
u_min=acos((X(2)^2+X(3)^2-(X(4)-X(1))^2)/(2*X(2)*X(3)));
u_max=acos((X(2)^2+X(3)^2-(X(4)+X(1))^2)/(2*X(2)*X(3)));


CV3=0;
CV4=0;
for i1=1:6
```

```
   nume=X(1)^2+X(4)^2-X(2)^2-X(3)^2-2*X(1)*X(4)*cos(X(i1+6));
   deno=(-2*X(2)*X(3));
   u(i1)=acos(nume/deno);
   g2=(u_min/u(i1))-1;
    g3=(u(i1)/u_min)-1;
     if g2>0
         CV3_P=g2;
     else
         CV3_P=0;
     end
     if g3>0
         CV4_P=g3;
     else
         CV4_P=0;
     end
     CV3=CV3+CV3_P;
     CV4=CV4+CV4_P;
end
const=1000*(CV1+CV2);%+CV3+CV4);

k1=X(4)/X(1);k2=X(4)/X(3);k3=(X(1)^2+X(3)^2+X(4)^2-X(2)^2)/(2*X(1)*X(3));
k4=X(4)/X(2);k5=(-X(1)^2+X(3)^2-X(4)^2-X(2)^2)/(2*X(1)*X(2));
R=[cos(X(13)) -sin(X(13));sin(X(13)) cos(X(13))];
         T=[X(14);X(15)];
     j2=6;
     while j2<12
        E=-2*sin(X(j2+1));D=cos(X(j2+1))-k1+k4*cos(X(j2+1))+k5;
        F=k1+(k4-1)*cos(X(j2+1))+k5;i1=j2+1;
        if E^2-4*D*F<0
          px1=1e10;py1=1e10;px2=1e10;py2=1e10;
        else
           theta_31=2*atan((-E+sqrt(E^2-4*D*F))/(2*D));
           theta_32=2*atan((-E-sqrt(E^2-4*D*F))/(2*D));
           px1=X(1)*cos(X(i1))+X(5)*cos(theta_31)-X(6)*sin(theta_31);
           py1=X(1)*sin(X(i1))+X(5)*sin(theta_31)+X(6)*cos(theta_31);
           px2=X(1)*cos(X(i1))+X(5)*cos(theta_32)-X(6)*sin(theta_32);
           py2=X(1)*sin(X(i1))+X(5)*sin(theta_32)+X(6)*cos(theta_32);
        end


         P1=[px1;py1]; P2=[px2;py2];
         P01=R*P1+T;P02=R*P2+T;
         px01=P01(1);py01=P01(2);px02=P02(1);py02=P02(2);
        E1=(pxd(i1-6)-px01)^2+(pyd(i1-6)-py01)^2;
        E2=(pxd(i1-6)-px02)^2+(pyd(i1-6)-py02)^2;
        Er=min([E1 E2]);
        error=error+Er;
         j2=j2+1;
      end
  obj=error;
```